

## **PROGRAMA PARA CREAR MODELOS VIRTUALES DE EDIFICACIONES (PROYECTO 15%)**

**Objetivo:** Sean tan creativos como puedan, disfruten el proyecto y apliquen al máximo lo aprendido en clase.

**Entregables físicos:** Código, versión ejecutable y presentación usada en la exposición

**Exposición:** (15 min) 5 min por persona. La exposición del proyecto incluye una explicación general de la estructura del programa y diagrama UML, una muestra de una edificación creada de nivel de complejidad medio (más de 30 elementos), una muestra en tiempo real del funcionamiento del programa (como un usuario). La exposición es la muestra de lo que alcanzaron y por tanto es igual de importante que un buen proyecto.

### **Guías:**

#### **- Grupos de 3 personas**

- Usar programación orientada a objetos y crear un diagrama UML del proyecto.
- Cree una API y una GUI (basada en lo desarrollado en la API).
- La unidad de trabajo es mm
- Para cada elemento (vigas, columnas, piso, pared, ventana, puerta) definir un color.

**- Copia de proyectos entre grupos implica anulación de los proyectos implicados (tanto el original como el copiado) más las sanciones estipuladas en el reglamento.**

**- Realícelo con los lenguajes y librerías que desee.**

## **REQUERIMIENTOS DE USUARIO DESDE LA GUI**

- 1) **Al iniciar el programa grafique un sistema de referencia Global.**
- 2) Pida al usuario que defina las dimensiones de una “caja” que delimite la zona donde estará la edificación. Pida al usuario un numero de divisiones en x,y,z y genere y grafique un arreglo (“malla”) cartesiano de puntos en 3D.
- 3) El usuario ira creando la edificación elemento a elemento de la siguiente manera:
  - 3.1 Selecciona un tipo de elemento (vigas, columnas, piso, pared, ventana, puerta) que desea crear
  - 3.2 Selecciona puntos de la malla que ayudan a definir cada elemento (según el caso tienen diferente significado los puntos, e.g. en una viga los 2 primeros definen la longitud y el 3ero la orientación de la sección)
  - 3.3 Define los parámetros faltantes del elemento a crear (e.g. espesor, diámetro, base y altura sección) y selecciona crear.
  - 3.4 Tras hacer clic en crear, el objeto debe ser graficado con su sistema de referencia local.
- 4) El usuario debe tener forma de ir viendo lo creado en 3D es decir que pueda explorar su modelo viéndolo de diferentes puntos y vistas

- 5) Debe haber un botón y una función que cree una vista nueva donde se generen las proyecciones ortogonales de la construcción teniendo en cuenta solo vigas, columnas, paredes y pisos. Recomendación: Multiplique la matriz geométrica por una matriz de transformación que proyecte los puntos al plano. Después grafique, en un fondo blanco, las líneas que delimitan los elementos.

## RECOMENDACIONES API Y ESTRUCTURA DEL PROGRAMA

- Crear una clase abstracta que es la superclase de las siguientes clases (leer abajo), que contenga como atributos una matriz topológica y una geométrica, y como métodos una función que grafique los objetos basados en la matriz topológica y en la geométrica (Recordar actividades taller STL)
- Crear 3 subclases: una de elementos “1D” (para viga y columna), una de elementos “2D” (para paredes, pisos) y una para elementos más complejos (Puertas y ventanas).
- Cada subclase debe tener una función constructora (función que se llama cuando se inicializa el objeto). Según la clase, esta se encargará de crear la matriz geométrica y topológica del objeto basado en su versión “estándar” (que son una matriz geométrica y topológica de acceso global para todos los objetos) y en los parámetros de entrada. **La matriz geométrica se creará a partir de la “estándar” utilizando TRANSFORMACIONES GEOMÉTRICAS.**

### Parámetros de entrada:

- Los parámetros de entrada de las vigas (o columnas) son:

**Tipo de Sección** (Rectangular o circular). Si es rectangular, **Base y Altura**. Si es circular, **diámetro**.

**3 puntos** que se leen de la malla de puntos mencionada anteriormente. Dos de los puntos son utilizados para definir el inicio y fin de la viga; y la dirección del eje x local en el sistema de referencia global. El tercero es para definir la orientación del eje y local en el sistema de referencia global y por tanto la dirección de la sección transversal.

- Los parámetros de entrada de los pisos, y paredes son:

**Espesor**

**3 puntos** que definen la ubicación y orientación del sistema local del hexaedro y las dimensiones de la pared o el piso

- Los parámetros de entrada de las puertas y ventanas son (muy similares al anterior):

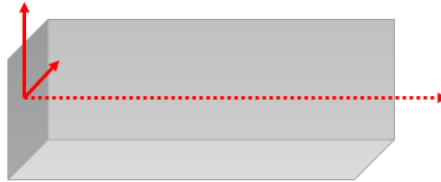
**Factor de escala en z local**

**3 puntos** que definen la ubicación y orientación del sistema local del hexaedro y las dimensiones de la pared o el piso

### Modelos “estándar”:

- La versión “estándar” de las vigas (o columnas) son:

Sección Rectangular: Base (Ylocal) y altura(Zlocal) de 1mm x 1mm y longitud de 1mm (Eje X local)



Sección “circular”: Diámetro de 1mm (Plano YZ local) y longitud de 1 mm (Eje x local)  
(Representéla como un Octágono)

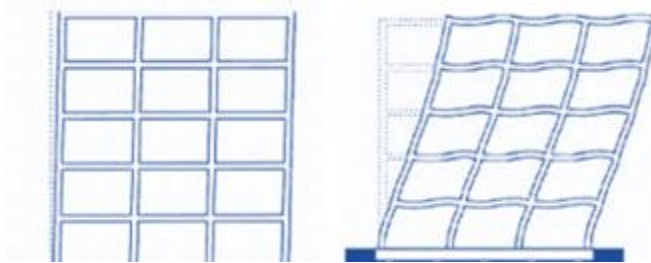
- La versión “estándar” de los pisos, y paredes es:

Hexaedro: Longitud 1 (Eje X local) de 1mm, Longitud 2 (Y local) de 1mm y espesor de 1mm espesor (Z local)

- Para la versión “estándar” de las puertas y ventanas busquen un modelo STL base, idealmente no con muchos triángulos. Y construyan una matriz topológica y geométrica base.

### Opcionales (Bonus)

- 1) Crear la posibilidad de guardar y cargar un proyecto por medio de la lectura y la escritura de un archivo de texto que contenga los elementos que definen la estructura.
- 2) Cree una “Morphing box” que anime la estructura después de un terremoto i.e. La base se mantiene fija y el resto se deforma como “un trampolín puesto vertical” o como “un árbol cuando hay viento”.



Nota: Un buen proyecto + la realización de opcionales (al menos 1) puede eximirlos del segundo proyecto o ayudarles con puntos para el parcial. El bonus específico es decisión del profesor con base en el trabajo realizado por el grupo de estudiantes.